Deep Head Pose Estimation from Depth Data for In-car Automotive Applications

Marco Venturelli, Guido Borghi, Roberto Vezzani, Rita Cucchiara

DIEF - University of Modena and Reggio Emilia
Via P. Vivarelli 10, 41125 Modena, Italy
Email: {name.surname}@unimore.it

Abstract. Recently, deep learning approaches have achieved promising results in various fields of computer vision. In this paper, we tackle the problem of head pose estimation through a Convolutional Neural Network (CNN). Differently from other proposals in the literature, the described system is able to work directly and based only on raw depth data. Moreover, the head pose estimation is solved as a regression problem and does not rely on visual facial features like facial landmarks. We tested our system on a well known public dataset, *Biwi Kinect Head Pose*, showing that our approach achieves state-of-art results and is able to meet real time performance requirements.

1 INTRODUCTION

Head pose estimation is an important visual cue in many fields, such as human intention, motivation, attention and so on. In particular, in automotive context, head pose estimation is one of the key elements for attention monitoring and driver behavior analysis.

Distracting driving has a crucial role in road crashes, as reported by the official US government website about distracted driving [1]. In particular, 18% of injury crashes were caused by distraction, more than 3000 people were killed in 2011 in a crash involving a distracted driver, and distraction is responsible for 11% of fatal crashes of drivers under the age of twenty [1]. The *National Safety Administration* (NHTSA) defines driving distraction as "*an activity that could divert a person's attention away from the primary task of driving*".

Driving distractions have been classified into three main categories [2]:

- **Manual Distraction**: the hands of the driver are not on the wheel; examples of this kind of activity are text messaging or incorrect use of infotainment system (radio, GPS navigation device and others).
- **Visual Distraction**: the driver does not look at the road, but, for example, at the smart-phone screen or a newspaper.
- **Cognitive Distraction**: the driver is not focused on driving activity; this could occur if talking with passengers or due to bad physical conditions (torpor, stress).

It is intuitive that smartphone is one of the most important cause of fatal driving distraction: it involves all three distraction categories mentioned above and it represents about 18% of fatal driver accidents in North America.
The introduction of semi-autonomous and autonomous driving vehicles and their coexistence with traditional cars is going to increase the already high interest about driver attention studies. Very likely, automatic pilots and human drivers will share the control of the vehicles, and the first will need to call back the latter when needed. For example, the same situation is currently happening on airplanes. The monitoring of the driver attention level is a key-enabling factor in this case. In addition, legal implications will be raised[3].
Among the others, a correct estimation of the driver head pose is an important element to accomplish driver attention and behavior monitoring during driving activity. To this aim, the placement and the choice of the most suitable sensing device is crucial. In particular, the final system should be able to work on each weather condition, like shining sun and clouds, in addition to sunrises, sunsets, nights that could dramatically change the quality and the visual performance of acquired images. Infrared or, even better, depth cameras overcome classical RGB sensors in this respect (see Fig. 1).

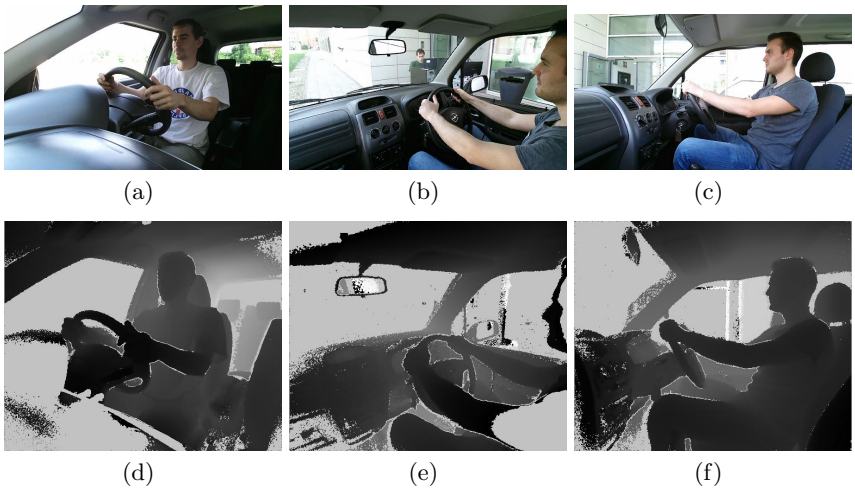

Fig. 1. Images acquired with *Microsoft Kinect One* device from different in-cockpit points of view. In the first row, RGB images are reported, while in the second row the corresponding depth maps are shown. It can be noted how light and the position of the camera could influence the view quality of the head and the other visible body parts and produce partial or severe occlusions.

In this work, we use and investigate the potentiality of depth images. The release of cheap but accurate 3D sensors, like *Microsoft Kinect*, brings up new

opportunities in this field and much more accurate depth maps. Existing depth-based methods either need manual or semi-automatic initialization, could not handle large pose variations and does not work in real time; all these elements are not admissible in automotive context.
Here, we propose an efficient and accurate head localization framework, exploiting Convolutional Neural Networks (CNN) in a data regression manner. The provided results confirm that a low quality depth input image is enough to achieve good performance. Although the recent advantages in classification tasks using CNNs, the lack of research on deep approaches for angle regression proves the complexity of this kind of task.

2 RELATED WORK

Head localization and pose estimation are the goal of several works in the literature [4]. Existing methods can be divided depending on the type of data they rely on: 2D (RGB or gray scale), 3D (RGB-D) data, or both. Due to the approach of our work, we briefly describe methods that use 2D data, and we focus on methods based on depth data or a combination of depth and intensity information. In general, methods relying solely on RGB images are sensitive to illumination, lack of features and partial occlusions [5].
To avoid these issues, [6] use for the first time Convolutional Neural Network to exploit CNN well-known power in space and color invariance. This is one of the first case in which a Convolutional Neural Network (CNN) is used in order to perform head pose estimation using images acquired by a monocular camera. This architecture is exploited in a data regression manner. A mapping function between three head predicted angles and visual appearance is learned. Despite the use of deep learning techniques, system is working in real time with the aid of a GPU. Also in [7] a CNN is exploited to predict head pose and gaze direction, a regression technique is approximated with a Softmax layer with 360 classes.
Besides, CNN is used in [8]: the network is trained on synthetic images. Recently, the use of synthetic dataset is increasing to support deep learning approaches that basically require huge amount of data. In [9] the problem of head pose estimation is taken on extremely low resolution images, achieving results very close to state-of-art results for full resolution images. [10] used HOG features and a Gaussian locally-linear mapping model, learned using training data, to map the face descriptor onto the space of head poses and to predict angles of head rotation.
Malassiotis et al. in [11] proposed a method based on low quality depth data to perform head localization and pose estimation; this method relies on the accurate localization of the nose and that could be a strong limitation for automotive context. Bretenstein et al. in [12] proposed a real time method which can handle large pose variation, partial occlusion and facial expression from range images; the main issue is that the nose must to be always visible, this method uses geometric features to generate nose candidates which suggest head position hy-

pothesis. The alignment error computation is demanded to a dedicated GPU in order to work in real time.
[13] investigated an algorithm based on least-square minimization of the difference between the measured rate of change of depth at a point and the rate predicted, to perform head localization and then head detection and tracking during videos. Fanelli et al. in [5] proposed a real time framework based on Random Regression Forests to perform head pose estimation from depth images.
In [14] the head pose estimation is treated as a optimization problem that is solved through Particle Swarm Optimization. This method needs a a frame (the first of the sequence) to construct the reference head pose from depth data; low real time performance are obtained thanks to a GPU. Papazov et al. in [15] introduced a novel triangular surface patch descriptor to encode shapes of 3D surface; this descriptor of an input depth map is matched to the most similar ones that were computed from synthetic head models in a training phase.
Seeman et al. in [16] proposed a method based on neural network and a combination of depth information, acquired by a stereo camera, and skin color histograms derived from RGB images; in this case work limit is that the user face has to be detected in frontal pose at the beginning of framework pipeline. [17] presented a solution for real time head pose estimation based on the fusion of color and time-of-flight depth data. The computation work is demanded to a dedicated GPU.
Baltrusaitis et al. in [18] presented a 3D constrained local method for robust facial feature tracking under varying poses, based on the integration both depth and intensity information. In this case the head pose estimation is one of the consequences of landmark tracking. A method to elaborate HOG features both on 2D (intensity) and depth data is described in [19, 20]; in the first case a Multi Layer Perceptron is the used for classification task; in the second, a SVM is used.
Ghiass et al. [21] performed pose estimation by fitting a 3D morphable model which included pose parameter, starting both from RGB and depth data. This method relies on detector of Viola and Jones [22].

3 HEAD POSE ESTIMATION

The goal of the system is the estimation of the head pose (i.e., pitch, roll, yaw angles with respect to a frontal pose) directly from depth data using a deep learning approach. We suppose to have a correct head detection and localization. The description of these steps are out of the scope of this paper. Differently from [16, 11, 12], additional information such as facial landmarks, nose tip position, skin color and so on are not taken into account.

3.1 Image pre-processing

Image pre-processing is a fundamental step to obtain better performance with the further exploitation of CNN [23].
First of all, the face images are cropped using a dynamic window. Given the

center x_c, y_c of the face, the image is cropped to a rectangular box centered in x_c, y_c with width and height computed as:

$$w, h = \frac{f_{x,y} R}{Z},$$

where $f_{x,y}$ are the horizontal and vertical focal lengths (in pixels) of the acquisition device, R is the width of a generic face (120 mm in our experiments, [6]) and Z is the distance between the acquisition device and the user obtained from the depth image. The output is an image which contains very few parts of background. Then, the cropped images are resized to 64x64 pixels and their values are normalized so that the mean and the variance are 0 and 1, respectively. This normalization is also required by the specific activation function of the network layers (see Section 3.2). Finally, to further reduce the impact of the background pixels, each image row is linearly stretched (see Algorithm 1) keeping only foreground pixels. Some example results are reported in Figure 2.

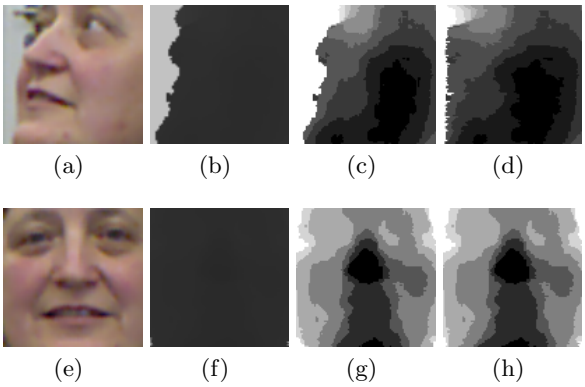

Fig. 2. Example of image pre-processing on two different input cropped image: (a) is the RGB frame, (b) the correspondent depth map, (c) depth map after normalization and (d) depth map after the linear interpolation. From (e) to (h) is the same, but with a frontal point of view. It can be noted that in (h) interpolation does not change a lot the visual result, due to the absence of background.

3.2 Deep Architecture

The architecture of the neural network is inspired from the one proposed by Ahn *et al.* [6]. We adopt a shallow deep architecture, in order to obtain a real time system and to maintain good accuracy. The network takes images of 64x64 pixels as input, which are relatively smaller than other deep architecture for face applications.
The proposed structure is depicted in Figure 3. It is composed of 5 convolutional

Algorithm 1 Linear Interpolation Algorithm

procedure LINEAR INTERPOLATION
  w : image width
  for row in image rows **do**
    x_{min} = first foreground pixel in row
    x_{max} = last foreground pixel in row
    for x=0 to w-1 **do**
      $x_{src} = x/(w-1) * (x_{max} - x_{min})$
      $x_1 = \lfloor x_{src} \rfloor$
      $x_2 = x_1 + 1$
      if $x_2 \leq w$ **then**
        $\lambda = x_2 - x_{src}$
        $row_{out}[x] = row[x_1] * \lambda + row[x_2] * (1 - \lambda)$
      else
        $row_{out}[x] = row[x_1]$

layers; the first four have 30 filters whereas the last one has 120 filters. At the end of the network there are three fully connected layers, with 120, 84 and 3 neurons respectively, that correspond to the three head angles (yaw, pitch and roll). The size of the convolution filters are 5x5, 4x4, 3x3, depending on the layer. Max-pooling is conducted only three times. The activation function is the hyperbolic tangent: in this way network can map output $[-\infty, +\infty] \rightarrow [-1, +1]$, even if ReLU tends to train faster that other activation functions [23]. In this way, the network outputs continuous instead of discrete values. We adopt the Stochastic Gradient Descent (SGD) as in [23] to resolve back-propagation.
A L2 loss function is exploited:

$$loss = \sum_{i}^{n} \|\mathbf{y_i} - \mathbf{f}(\mathbf{x_i})\|_2^2.$$

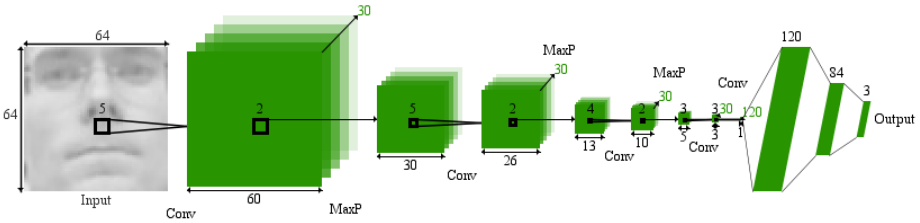

Fig. 3. The deep architecture that represents network adopted in our work: input is a 64x64 image, there are 5 convolutional layers, 3 fully connected layers; output has 3 neurons to predict head yaw, pitch and roll. This chart is obtained with *DeepVisualizer*.

The network representation in Figure 3 is obtained using *DeepVisualizer*, a software we recently developed. [1]

3.3 Training

The network has been trained with a batch size of 64, a decay value of 5^{-4}, a momentum value of 9^{-1} and a learning rate set to 10^{-1}, descending to 10^{-3} in the final epochs [23]. Ground truth angles are normalized to $[-1, +1]$.
An important and fundamental aspect of deep learning is the amount of training data. To this aim, we performed data augmentation to avoid over fitting on limited datasets. For each pre-processed input image, 10 additional images are generated. 5 patches are randomly cropped from each corner and from their center, other 4 patches are extracted by cropping original images starting from the bottom, upper, left and right image part. Finally, one more patch is created adding Gaussian noise (*jittering*).

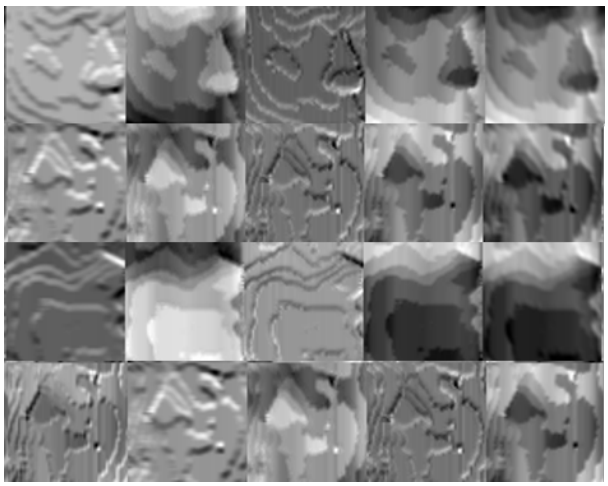

Fig. 4. Some example of feature maps generated by our network. The network has learned to extract facial elements like nose tip, eye holes, cheeks and contour lines.

[1] The tool is written in Java and it is completely free and open source. It takes as input the JSON file produced by the Keras framework and generates image outputs in common formats such as png, jpeg or gif. We invite the readers to test and use this software, hoping it can help in deep learning studies and presentations. The code can be downloaded at the following link:
`http://imagelab.ing.unimore.it/deepvisualizer`

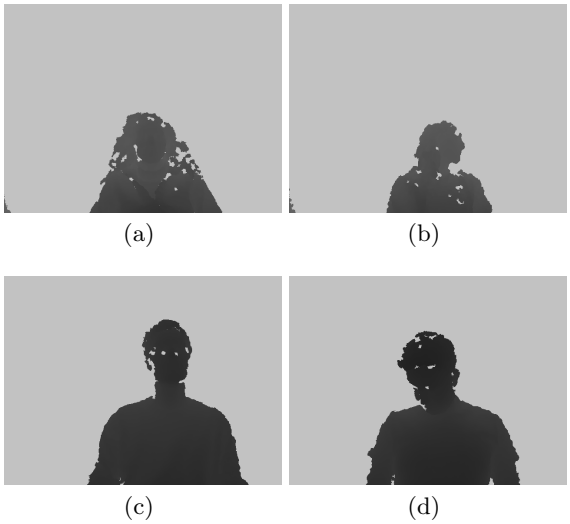

Fig. 5. Some example of *Biwi* dataset frames that present visual artifacts, like holes, with female (a - b) and male (c - d) subjects.

4 EXPERIMENTAL RESULTS

In order to evaluate the performance of the presented method, we use a public dataset for head pose estimation that contain both RGB and depth data, namely *Biwi Kinect Head Pose Database*.

4.1 Biwi Kinect Head Pose Database

This dataset is introduced by Fanelli *et al.* in [24] and it is explicitly designed for head pose estimation task. It contains 15678 upper body images of 20 people (14 males and 6 females) and 4 people were recorded twice. The head rotation spans about $\pm 75°$ for yaw, $\pm 60°$ for pitch and $\pm 50°$ for roll. For each frame a depth image and the corresponding RGB image are provided, acquired sitting in front a stationary *Microsoft Kinect*; both of them have a resolution of 640x480. Besides ground truth pose angles, calibration matrix and head center - the position of the nose tip - are given.
This is a challenging dataset because of the low quality of depth images (*e.g.* long hair in female subjects cause holes in depth maps, some subject wear glasses, see Figure 5); besides the total number of samples used for training and testing and the subjects selection are not clear, even in the original work [5]. To avoid this ambiguity, we use sequences 1 and 12 to test our network, which correspond to not repeated subjects. Some papers use own method to collect results (*e.g.* [6]), so their results are not reported and compared.

4.2 Discussion about other datasets

Several dataset for head pose estimation were collected in this decade, but in most cases there are some not desirable issues. The main issues are that they do not provide depth data (*e.g.*, *RobeSafe Driver Monitoring Video Dataset* [25]), or that not all angles are included (*e.g.*, *Florence 2D/3D Face Dataset* [26] reports only yaw angles). Moreover, most of the datasets have not enough frames or images for deep learning approaches.
ICT-3DHP Dataset [18] is collected using *Microsoft Kinect* sensor. It contains about 14000 frames both intensity and depth. The ground truth is labeled using a *Polhemus Fastrack* flock of birds tracker. This dataset has three main drawbacks: users had to wear a white cap for the tracking system. The cap is well visible both in RGB and depth video sequences. Second, there is a lack of training data images with roll angles and the head center position is not so accurate (see Figure 6). Finally, this dataset is not good for deep learning, because of its small size and the presence of few subjects.

(a) (b)

Fig. 6. Two frames of the *ICT-3DHP Dataset*. At the right can be seen the white cap, at the left the correspondent head center position that is translated to the left.

4.3 Quantitative results

Table 1 reports the results obtained on *Biwi Kinect Head Pose Dataset*. We follow the evaluation protocol proposed in [5]. Processing time is tested on *Nvidia Quadro k2200* 4GB GPU with the same test sequences.
Results reported in Table 1 show that our method overcomes other state-of-the-art techniques, even those working on both RGB and depth data or are based on deep learning approaches [7].
Thanks to the high accuracy reached, the proposed network can be used for efficient and precise head orientations applications, also in automotive context, with an impressively low elaboration time.
Figure 8 shows an example of working framework for head pose estimation in real time: head center is taken thanks to ground truth data; the face is cropped

Table 1. Results on *Biwi* dataset (Euler angles)

Met.	Data	Pitch	Roll	Yaw	Time
[20]	RGB+depth	5.0 ± 5.8	4.3 ± 4.6	3.9 ± 4.2	-
[7]	RGB+depth	4.76	-	5.32	-
[5]	depth	8.5 ± 9.9	7.9 ± 8.3	8.9 ± 13.0	40 ms/frame
[19]	RGB+depth	9.1 ± 7.4	7.4 ± 4.9	8.9 ± 8.2	100 ms/frame
[18]	RGB+depth	5.1	11.2	6.29	-
[15]	depth	3.0 ± 9.6	2.5 ± 7.4	3.8 ± 16.0	76 ms/frame
Our	depth	**2.8 ± 3.1**	**2.3 ± 2.9**	**3.6 ± 4.1**	**10 ms/frame**

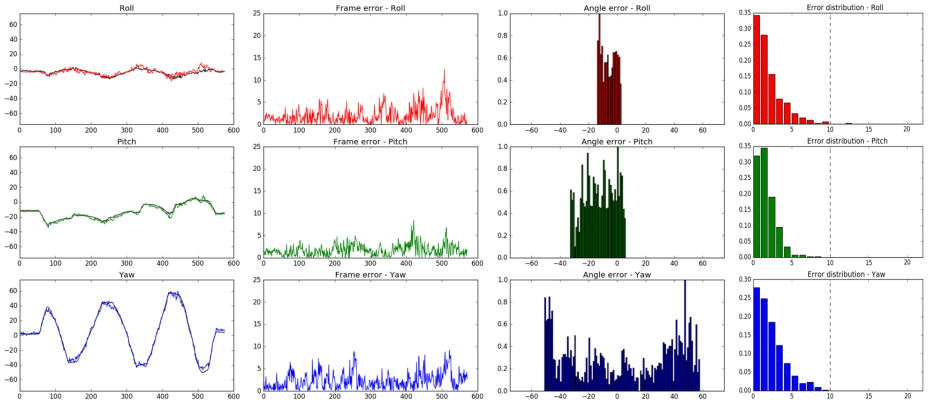

Fig. 7. Experimental results: roll, pitch and yaw angles are reported on the three rows. The ground truth is superimposed in black. The angle error per frame is reported in the second column, while in the third column histograms highlights the errors at specific angles. The error distribution is reported in the last column.

from raw depth map (in the center image, the blue rectangle) and in the right frame yaw, pitch and roll angles are shown.

5 CONCLUSIONS

We present a innovative method to directly extract head angles from depth images in real time, exploiting a deep learning approach. Our technique aim to deal with two main issue of deep architectures in general, and CNNs in particular: the difficulty to solve regression problems and the traditional heavy computational load that compromises real time performance for deep architectures.
Our approach is based on Convolutional Neural Network with shallow deep architecture, to preserve time performance, and is designed to resolve a regression task.
There is rich possibility for extensions thanks to the flexibility of our approach: in future work we plan to integrate temporal coherence and stabilization in the deep learning architecture, maintaining real time performance, incorporate RGB

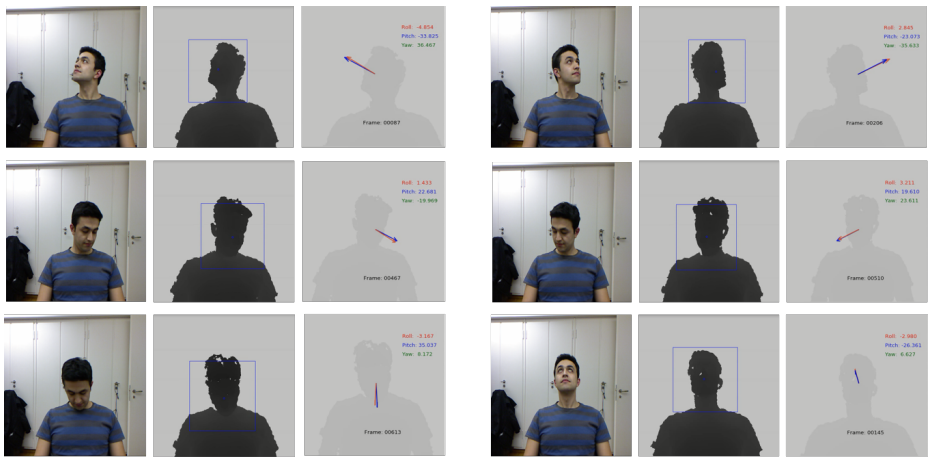

Fig. 8. The first column show RGB frames, the second the correspondent depth map frame: blue rectangle reveal the dynamic crop to extract the face. The last column reports yaw (red), pitch (blue) and roll (green) angles values and the frame number (*Biwi* dataset)

or infrared data to investigate the possibility to have a light invariant approach even in particular context (*e.g.* automotive). Head localization through deep approach could be studied in order to develop a complete framework that can detect, localize and estimate head pose inside a cockpit.
Besides studies about how occlusions can deprecate our method are being conducted.

References

1. "distraction.gov, official us government website for distracted driving," http://www.distraction.gov/index.html, accessed: 2016-09-01.
2. C. Craye and F. Karray, "Driver distraction detection and recognition using RGB-D sensor," *CoRR*, vol. abs/1502.00250, 2015. [Online]. Available: http://arxiv.org/abs/1502.00250
3. H. Rahman, S. Begum, and M. U. Ahmed, "Driver monitoring in the context of autonomous vehicle," November 2015. [Online]. Available: http://www.es.mdh.se/publications/4021-
4. E. Murphy-Chutorian and M. M. Trivedi, "Head pose estimation in computer vision: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 607–626, Apr. 2009. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2008.106
5. G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 617–624.
6. B. Ahn, J. Park, and I. S. Kweon, "Real-time head orientation from a monocular camera using deep neural network," in *Asian Conference on Computer Vision*. Springer, 2014, pp. 82–96.

7. S. S. Mukherjee and N. M. Robertson, "Deep head pose: Gaze-direction estimation in multimodal video," *IEEE Transactions on Multimedia*, vol. 17, no. 11, pp. 2094–2107, 2015.
8. X. Liu, W. Liang, Y. Wang, S. Li, and M. Pei, "3d head pose estimation with convolutional neural network trained on synthetic images," in *Image Processing (ICIP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 1289–1293.
9. J. Chen, J. Wu, K. Richter, J. Konrad, and P. Ishwar, "Estimating head pose orientation using extremely low resolution images," in *2016 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI)*. IEEE, 2016, pp. 65–68.
10. V. Drouard, S. Ba, G. Evangelidis, A. Deleforge, and R. Horaud, "Head pose estimation via probabilistic high-dimensional regression," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4624–4628.
11. S. Malassiotis and M. G. Strintzis, "Robust real-time 3d head pose estimation from range data," *Pattern Recognition*, vol. 38, no. 8, pp. 1153–1165, 2005.
12. M. D. Breitenstein, D. Kuettel, T. Weise, L. Van Gool, and H. Pfister, "Real-time face pose estimation from single range images," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*. IEEE, 2008, pp. 1–8.
13. F. A. Kondori, S. Yousefi, H. Li, S. Sonning, and S. Sonning, "3d head pose estimation using the kinect," in *Wireless Communications and Signal Processing (WCSP), 2011 International Conference on*. IEEE, 2011, pp. 1–4.
14. P. Padeleris, X. Zabulis, and A. A. Argyros, "Head pose estimation on depth data based on particle swarm optimization," in *2012 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*. IEEE, 2012, pp. 42–49.
15. C. Papazov, T. K. Marks, and M. Jones, "Real-time 3d head pose and facial landmark estimation from depth images using triangular surface patch features," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 4722–4730.
16. E. Seemann, K. Nickel, and R. Stiefelhagen, "Head pose estimation using stereo vision for human-robot interaction." in *FGR*. IEEE Computer Society, 2004, pp. 626–631. [Online]. Available: http://dblp.uni-trier.de/db/conf/fgr/fgr2004.html
17. A. Bleiweiss and M. Werman, "Robust head pose estimation by fusing time-of-flight depth and color," in *Multimedia Signal Processing (MMSP), 2010 IEEE International Workshop on*. IEEE, 2010, pp. 116–121.
18. T. Baltrušaitis, P. Robinson, and L.-P. Morency, "3d constrained local model for rigid and non-rigid facial tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2610–2617.
19. J. Yang, W. Liang, and Y. Jia, "Face pose estimation with combined 2d and 3d hog features," in *Pattern Recognition (ICPR), 2012 21st International Conference on*. IEEE, 2012, pp. 2492–2495.
20. A. Saeed and A. Al-Hamadi, "Boosted human head pose estimation using kinect camera," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1752–1756.
21. R. S. Ghiass, O. Arandjelović, and D. Laurendeau, "Highly accurate and fully automatic head pose estimation from a low quality consumer-level rgb-d sensor," in *Proceedings of the 2nd Workshop on Computational Models of Social Interactions: Human-Computer-Media Communication*. ACM, 2015, pp. 25–34.
22. P. Viola and M. J. Jones, "Robust real-time face detection," *International journal of computer vision*, vol. 57, no. 2, pp. 137–154, 2004.
23. A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

24. G. Fanelli, M. Dantone, J. Gall, A. Fossati, and L. Van Gool, "Random forests for real time 3d face analysis," *International Journal of Computer Vision*, vol. 101, no. 3, pp. 437–458, 2013.
25. J. Nuevo, L. M. Bergasa, and P. Jiménez, "Rsmat: Robust simultaneous modeling and tracking," *Pattern Recognition Letters*, vol. 31, pp. 2455–2463, December 2010. [Online]. Available: http://dx.doi.org/10.1016/j.patrec.2010.07.016
26. A. D. Bagdanov, I. Masi, and A. Del Bimbo, "The florence 2d/3d hybrid face datset," in *Proc. of ACM Multimedia Int.l Workshop on Multimedia access to 3D Human Objects (MA3HO11)*, ACM. ACM Press, December 2011.